

COSC-1420 C Programming

L02 Documentation

Fall 2014

L02 Documentation

Read this document, then read it again. Next, print it and keep it handy so you can refer to it during the entire course. You are expected to know this material and use it in your work.

Describe Documentation

Documentation tells us what was planned, what was done, and what, if any, fixes or changes have been applied or made.

Document (list or describe)

1. Document the purpose of the program or module
2. Document the main() function and each additional function
3. Document any change history for each module **for each change**, I expect to find three things:
 1. Date: when was the change made
 2. Problem: what was the problem that required a change
 3. Solution: what change was made to fix the problem

The change history should follow the intro comments at the top of the module that was changed and should come before any code for that module.

4. Document the input variables

- * name, data type, purpose, any initialization

5. Document working variables

- * name, data type, purpose, any initialization

6. Document the output variables

- * name, data type, purpose

7. Document Program Processing Steps

- * List the major steps needed to accomplish the task

8. Document Program Testing

- * What steps will be taken to demonstrate that the program works as intended?
- * Before the program is written, determine the test data that will be used and the expected results.

COSC-1420 C Programming

L02 Documentation

Fall 2014

- * Desk check your ideas prior to writing actual code.

9. Insert Comments as Needed

- * If the code is self-documenting, no additional comments may be needed.
- * Add comments to make the purpose of the code or block clear to a human reader.
- * Comments do not affect the size of the final executable, so be generous (but not excessive) with your comments.

Conventions

1. Every program must begin with a comment. The comment should include your:

- Name
- Section Number
- Lab or Program Number
- Date of Submission
- Brief description of the program
- Any revision history for the program

2. Variable names must be meaningful. The variable name `b` is usually not meaningful. The variable name `payrate` might be very meaningful.
3. Define each variable on a separate line. Consider adding a comment that describes the variable. If the name is self-describing, no further comment may be needed.
4. Include variable initialization when appropriate. If a particular initial value is important to the proper functioning of your program, then initialize the variable. This is true even if the initial value is also the system default value.
5. Variable names, in this course, are all lowercase with no spaces. This is a convention for this course. Other naming conventions exist but are not appropriate.
6. Insert horizontal spacing.
 - * Put a space character after each comma.
 - * Put a space before and after `+`, `-`, `*`, and `/`
 - * Use additional spaces to increase readability

7. Insert vertical spacing.

- * Add a blank line between the first comment and the `#` commands
- * Add a blank line between the `#` commands and function prototypes (if any)

COSC-1420 C Programming
L02 Documentation
Fall 2014

- * Insert a blank line between function prototypes (if any) and main
8. `main()` always returns void (for this course).
 9. Input consists of
 - * A comment
 - * Text (a prompt) that tells the user what is needed (typically `printf()`)
 - * A command that accepts the desired input (typically `scanf()`, sometime `gets()`)
 - * A blank line
 10. Processing - insert blank lines as needed to increase readability. Add specific comments to help the reader understand what is being done and why.
 11. Output consists of
 - * An appropriate comment
 - * Appropriate output statements (e.g., `printf()`)
 - * A blank line
 12. All blocks of code are surrounded by `{ }` (braces). This is true even if the block consists of a single line of code.
 13. The last output to the screen should always be followed by a NL (`'\n'`) character to move any system messages to the next line down on the screen.
 14. `main()` is followed by function definitions (if any)

Program Names

Your programs should be named as follows: FLNNNpXXXXrx.c Where

1. F is the first letter of your first name
2. L is the first letter of your last name
3. NNN are the last three digits of your Colleague ID
4. p is the lowercase character 'p'
5. XXXX is the lab (program) number (may be three or four characters). If the assignment is L10a, in the program name you would use p10a.
6. rx (the lowercase character 'r') is the revision level. For the initial version of your program this is left void (no character). After the first time you turn in a program, this becomes ra, rb, rc, etc. to indicate successive versions of the program.
7. If your name is Jane Franks with a Colleague ID of 123456 and you are turning in the assignment for program L10a, your first submission would have a filename of:

COSC-1420 C Programming
L02 Documentation
Fall 2014

Jf456p10a.c

If you need to revise your program, the next version would be named:

Jf456p10ara.c

Program Structure

1. Declare and initialize variables
2. Display any needed instructions
3. Get all needed inputs
4. Do all needed processing
5. Display all needed outputs
6. Unless essential to the functioning of the program, do not mix input, processing, and output

main(), and all other functions, always end with a return statement.

Function Headers

Every function must begin with a comment. The comment should include:

Function Name

Programmer Name

Date of Creation

Brief description of the function

Revision History (if needed)

Date: the date of the revision

Problem: what problem needed to be fixed

Solution: what change was made to fix the problem